PharmaSUG 2025 - Paper MM-384

The Model Maketh the Metadata

Carlo Radovsky, Immanant

ABSTRACT

Since well before the passage of FDASIA (FDA Safety and Innovation Act) in 2012 in the United States, which ultimately led to the US FDA requiring adherence to CDISC standards in the submission of clinical trials data, the biopharmaceutical industry has known what it had to do – find a way to streamline processes and solutions. For just as long, it has struggled with how to do so. The rallying cry has been for an automated end-to-end solution, which has proved to be an elusive unicorn, at best painfully, partially supported by commercial products and often stymied by internal silos and broad-based technical debt.

While there are a number of proximate causes for this, automating the generation of submission datasets begins with a metadata reference, be it the published standards documents, spreadsheets, or a robust repository. To date, the industry has looked to CDISC (Clinical Data Interchange Standards Consortium) materials as a starting point, quickly finding that, as published, these metadata sources are only a starting point both in content and model, leaving everyone to build proprietary solutions.

This paper proposes a more comprehensive representation, establishing a robust model that supports both internal standards and study variability, while also streamlining version management, up-versioning, and end-to-end processes.

INTRODUCTION

A small spring can be found on the slopes of Mount Helicon in Greece. Legend says it was formed when Pegasus tamed the mountain's exuberance by striking it with his hoof, quieting an earthquake and creating one of the primordial wells of inspiration.

Figure 1. Pegasus at Hippocrene

Truth be told, the actual spring is underwhelming. Humble in appearance, it is a gap in the rocks not much larger than a bucket to be lowered into it. In no way does it reflect the awesome power of poetry's muse supposedly bestowed on those who drink its blushful, maddening draughts.

The CDISC Library is a similar wellspring: a humble, single source of truth that could be considered baseline metadata for data standards and targeted implementations (e.g., the Tobacco Implementation Guide or TIG). It also contains a meaningful subset of CDISC's published terminology standards. (Note that, more recently, with accessing the library via API calls, users can also extract content related to biomedical concepts, but that is not in scope for this paper).

But what is the Library, exactly? How is this content structured and how does it serve industry? Is there perhaps a better approach to take in modeling the content, such that consumers are both more aligned with each other, and have an easier time of operationalizing it? And, how in the world can versioning over time be managed as implemented both an organization's bespoke standards, as well as for individual studies?

Before answering the latter questions, a first look at the first ones. Also, while the examples used are primarily related to SDTM, the identified challenges are broadly applicable to all data standards models.

DATA STANDARDS METADATA – DIGITIZED NOT DIGITALIZED

The metadata for data standards (CDASH model, SDTMIG, ADaMIG, etc.) available from the CDISC Library is largely based on specification tables as published in documents such as the SDTM model and ADaM Implementation Guide. In the early years, documents were the only reference materials available to industry. This evolved over time, first to a spreadsheet format then to an actual reference repository, first named SHARE (Shared Health and Research Environment), and now the CDISC Library that is available today.

This is an exemplar of digitization: what you see in the document is what you get in the Library, with a bit of a structural wrapper. In other words, the model structure is primarily determined by the document organization of content, making it the source of truth for both the metadata model as well as the content. This leads to some significant constraints, making it not so much a baseline as a starting point – or many starting points – that have to be pre-processed to be effectively consumed and operationalized. While this may be understandable given the historical evolution, it has led to modeling that is optimized for publication rather than consumption.

A couple of items to illustrate this.

SDTM MODELING – TEN-THOUSAND FOOT VIEW

First, a look at the model content for SDTM in the CDISC Library reveals a hierarchical structure with Classes identified under the Model:

Figure 2. Hierarchy of Top Levels in the SDTM v2.0 Model

General Observations and ***Associated Persons***, however, are not classes. As published, they identify variable groupings applicable to certain of the "other" classes defined in the model. They likely are identified as classes because they are distinct sections within the model document. Which classes they apply to, and how these variable groups are integrated into a class as a whole, are detailed in the document, and in the descriptions and other exposition in the Library. From a data model perspective, however, these considerations have to be implicitly understood by consumers of the Library content. Additionally, ***General Observations*** is a placeholder to capture those variables that are published in tables listing Identifier and Timing variables for all classes. There is a conceptual mis-alignment, in that the variables in these tables can apply to domains beyond the ***General Observation*** class (e.g., Timing variables allowed in ***Special Purpose*** domains).

Findings About (FA) is a bit of an orphaned concept in SDTM. In many ways, it is like ***Associated Persons***, a variable group supporting the specialization of the Findings class. In the published terminology supporting Define-XML, FA is listed as one of the primary classes on par with ***Findings*** (unlike ***General Observations*** and ***Associated Persons***, which are not). In the latest SDTM Model document, FA is termed a "specialization", while the latest SDTMIG document somewhat casually identifies FA as a "subclass", a concept used in other CDISC standards, but not formally established in SDTM.

SDTM MODELING – TEN-FOOT VIEW

At a more granular level, consider the model-permissible variables --OCCUR, --STAT, and --REASND in relation to the Adverse Events (AE) domain. These variables are generally allowed in ***Events*** domains (model permissible), but at least since SDTMIG v3.1.2, it has been explicitly stated that they are not allowed in AE (SDTMIG v3.1.2, Section 6.2.1.1, Assumption 8). The prohibition is also present in the subsequent versions of the SDTMIG documents.

For SDTMIGs 3.2, 3.3, and 3.4, the Library contains no indication of this constraint on the AE Domain. Its only representation is in the SDTM 2.0 metadata, and that is due to an expanded metadata specification table format in the SDTM 2.0 document.

This, ultimately, highlights the problem. While the prohibition is consistent in the standard, it is not only generally absent from metadata, it differs over time (see the below table).

Source & Version	Document reference of Prohibition	Prohibition Supported by CDISC Library
SDTMIG v3.1.2	Section 6.2.1.1, Assumption 8	No
SDTM v 1.2	None	No
SDTMIG v3.2	Section 6.2, AE Section, Assumption 8	No
SDTM v 1.4	None	No
SDTMIG v3.3	Section 6.2.1, Assumption 9	No
SDTM v 1.7	None	No
SDTMIG v3.4	Section 6.2.1, Assumption 10	No
SDTM v 2.0	Section 3.1.2, Table	Yes

Table 1. Prohibition of --OCCUR, --STAT, and --REASND in AE Domain

In addition, the prohibition is identified only in the model metadata, and not in the metadata for the associated IG. In other words, for end users that might reference only the IG Library content, there would be no indication that model-permissible variables such as –STAT are prohibited from specific domains. Lastly, when it is represented in the library, the prohibition is embedded in semi-structured text, not an explicit attribute with defined requirements (i.e., not really data).

TERMINOLOGY REFERENCES - ORPHANED IN TIME

The terminology references are a somewhat different kettle of sea life. They have been published in much the same format for almost twenty years. But while the format is familiar, it may be that familiarity has bred misunderstanding.

At the root of the misunderstanding is the C-Code. The C-Code is often mischaracterized as a CDISC Code, when it is in fact a Concept Code as established and maintained by the US National Cancer Institute (NCI). It is a single-value, unique identifier of the ***concept*** associated with a codelist (a finite set of allowable values) or the ***concept*** associated with a term (an individual allowed value in a codelist), as defined by NCI.

Many organizations have treated the C-Code as a single-value primary key related to terminology, a unique identifier of a codelist or term. Not only is this problematic, but formally speaking, it is incorrect to view the C-Code this way.

In terms of the C-Codes associated with codelists, they are a fairly reliable unique identifier. There are several reasons for this, but primarily it is that, for submission, the specific name of a codelist is less critical than the terms in that codelist.

As an example, the Define-XML codelist enumerating domain/dataset classes has a C-Code of C103329 and is defined as “Terminology related to the classification of a CDISC domain.” It contains all valid class values for SDTM, SEND, and ADaM datasets/domains. It is, however, named “General Observation Class”, with a short name of “GNRLOBSC”. These labels are both hold-overs from when the codelist was first established under the SDTM terminology, before being moved under Define-XML. They aren’t really aligned to the content or the concept of the current codelist. Changing the name and/or the C-Code might be appropriate from a purist standpoint, but has little operational benefit at the cost of backward compatibility.

For terms in a codelist, however, there has been a small, but meaningful destabilization of the C-Code as a unique identifier. This is because the relationship between concept and term is more likely to evolve over time. This evolution can be either orthographic (the term for submission can change, e.g., “UNK” might be expanded to “UNKNOWN”) or conceptual (the term is correct, but the concept associated with the term has been changed).

An example of this occurred in Package 58 of the SDTM terminology: the term “Indication” in the DOTEST codelist had its C-Code changed from “C41184” to “C112038” based on the alignment of the sematic meaning with a term in the TSPARM codelist.

More fundamentally, however, this leads to a disconnect between the NCI framework of concept codes and the operational use of terminology. It explains why, from the perspective of the NCI Enterprise Vocabulary Services (EVS), the term “BMI” in the Vital Signs Test Code codelist has the same C-Code (C16358) as the term “Body Mass Index” in the Vital Signs Test Name codelist – they are the same concept represented by two distinct terms. It also explains why, when a term is impacted by a change to the underlying concept, the C-Code changes as well.

This matters, in that you cannot query a terminology reference for the C-Code C16358 and get a single value back, nor be assured it identifies the same term over time, when it is the term that you have to manage operationally.

So then, the unique identifier for terms is a composite key, consisting of, well, what? At first glance, it could be the paired C-Codes of the codelist and term. But the C-Code can change over time while the orthographic term stays consistent. And the orthographic term can change over time while the C-Code stays consistent.

In other words, CDISC doesn’t publish or maintain a constant identifier over time. How do consumers manage this content? How can you tell when a term has been removed completely versus retired and reintroduced? How can you track terms that split or join over time? For the most part the answers are: not very well, with significant effort, and many often give up.

And that is the crux of industry’s challenge with terminology — each published version has a nominal change history and links to prior versions, but the information is largely ahistorical, with each published version existing as a stand-alone island, unconnected in a sea of other versions. The change log as published amounts to an oral history of what has gone on before, but this is at best actionable by subject-matter experts, and far from automatable.

IMPACT OF THESE LIMITATIONS

To say at the outset, these are not insurmountable problems. Organizations large and small have established robust processes and solutions for ingesting CDISC Library content.

But not without significant cost and effort. Spreadsheets are ever-expanding for those organizations without a metadata repository. For those with it, the modeling can become fractally complex, and content management reliant on a dedicated strike time on par with Special Forces in terms of technical and subject-matter expertise.

Across a number of organizations, going from the content model as represented in the CDISC Library to an operational model takes three to five transformation steps depending on the organization. These include steps such as:

- Original acquisition (i.e., reading in from source, whether spreadsheet, API, etc.)
- Content Impact analysis (changes against prior acquisition)
- Mapping to the operational model
- Up-versioning in the operational model
- Operational Impact analysis and integration into current use

While several of these steps may be more difficult due to the current metadata model, these challenges won’t go away just by refactoring the CDISC Library. But there are broad impacts of the current approach that could be lessened:

- Consumption of the metadata at is stands requires pre-processing to be both accurate and complete, with extensive reliance on subject-matter experts coordinating with implementers

- The resulting operational metadata model is unnecessarily proprietary. As the content moves through the process, each organization makes bespoke decisions on modeling based on their tools, processes, and use cases.
- Versioning and impact analysis are highly dependent upon the preprocessing algorithms and their assumptions.

From the perspective of end-to-end automation, all of this undercuts:

- Interoperability
- Vendor capabilities and solutions
- Industry adoption
- Industry efforts at collaboration and proof of concept (e.g., CDISC 360 efforts)

Put another way, the current metadata representation in the CDISC Library is at best a speedbump in automation, one that increasingly is becoming an obstacle for industry.

REMODEL & RENOVATE

First and foremost, the models as represented in the source documents have to be fully digitalized. The standards as published still remain as the conceptual source of truth. In contrast, the CDISC Library can become the operational metadata source of truth. This allows the model to go beyond the explicit organization and content in the documents while still adhering to a data standard's requirements.

As a starting point, refactor the metadata model with the following general principles in mind:

- Be consistent. If a modeling concept is applicable to a given version of a standard, then represent it consistently in the model across all versions of the standard to which it applies (e.g., prohibition of variables in a domain). Similarly, harmonize modeling concepts across standards. While the teams supporting CDISC data standards may feel strongly about standard-specific framing, implementers care far more about consistent and streamlined models that facilitate consistent and streamlined solutions.
- Disambiguate modeling concepts, attributes, and terminology (e.g., classes are aligned to published terminology)
- Restrict nodes in the model hierarchy to discrete objects. For example, ***Class, Dataset,*** and ***Variable*** are all objects that fully define something. In contrast, ***Variable Groups*** or ***Variable Sets*** identify sub-groups of variables in a ***Data Structure*** that require more than one to define a complete structure. They do not identify stand-alone structures.
- Model content with the end in mind. The goal should be to enable automation and interoperability. This requires a model that supports full traceability from one version to the next.
- Version content at all levels within the model. Keep versioning simple (e.g., any change, whatsoever, is a new version).

DATA STANDARD MODEL OVERVIEW

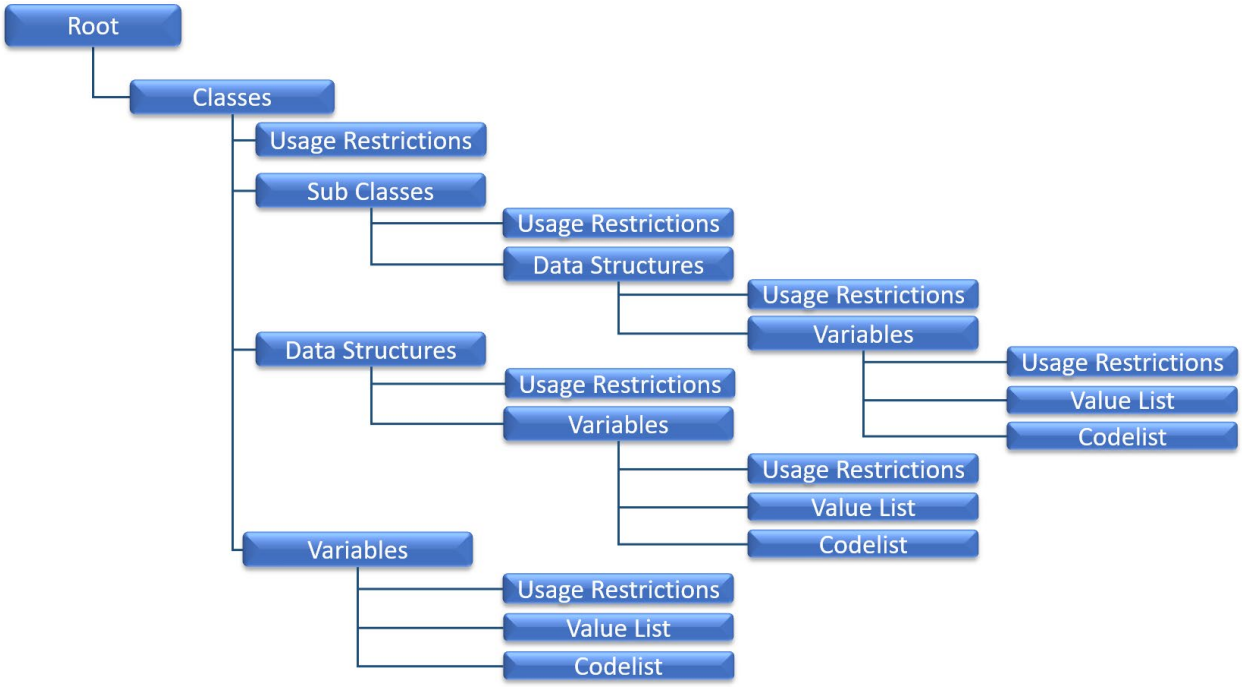

Figure 3. Data Standard Model Overview

The above represents an overview of the refactored model. It introduces ***Sub Classes*** as a distinct node, harmonizes concepts such as ***Domain*** and ***Dataset*** to ***Data Structure***, and establishes a clear hierarchy supporting all standards. It also introduces the ***Usage Restrictions*** node, discussed later.

One consideration to note: structurally, there is not much that distinguishes ***Classes, Sub Classes,*** and ***Dataset Structures***. They are all containers that capture either sub-containers or variables. The above could be streamlined further by establishing a container type with these terms. That certainly would be the most flexible model, allowing endlessly ramifying layers as needed. Conceptually, however, it makes little difference, and for ease of reference the node names above are used.

At first blush, this model could also apply to Define-XML-centric content, which terms ***Data Structures*** as ***ItemGroups*** and ***Variables*** as ***Items.*** In Define-XML, however, there are fundamental differences in the model hierarchy (e.g., ***Class*** is a direct attribute to under ***ItemGroup,*** rather than a node above it). Given how ***Class*** can contain both ***Variables*** as well as ***Sub Classes*** within the data standards, there is little benefit in trying to force harmonization.

Each node will be looked at step-by-step, detailing how the principles identified previously have been applied. Note that the modeling is conceptual and uses simple terms to keep it accessible. Similarly, while generally aligned to the model from the CDISC Library, it has been simplified for ease of discussion.

Lastly, there are implementation considerations (e.g., excessive re-querying or traversing depending on the underlying metadata architecture, versioning mechanics) that are dependent on the technology used to maintain the content and go beyond the scope of this paper.

Data Standard Root

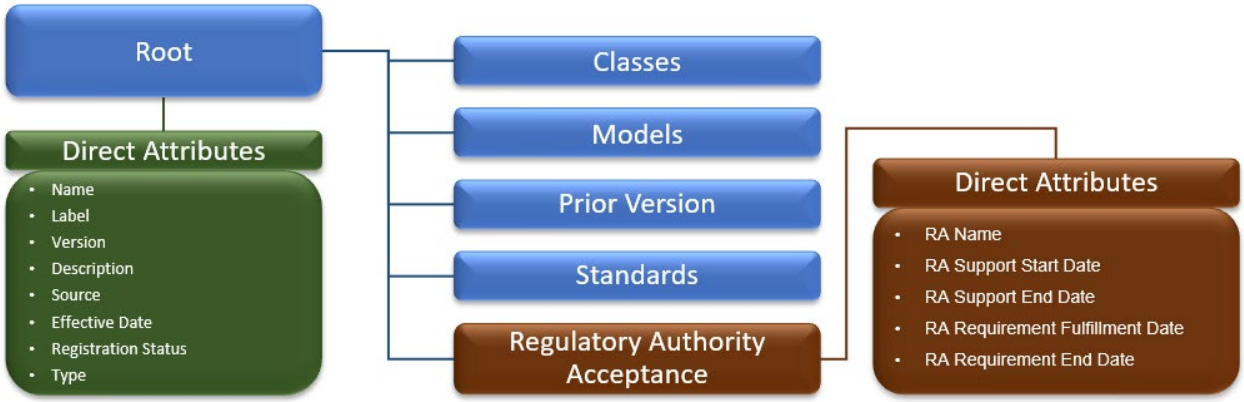

Figure 4 Data Standard Root Representation

This is a complete representation of the root node across all data standards, covering data standard models (SDTM, CDASH, etc.), IGs (ADaMIG, SENDIG, etc.) and integrated standards (TIG, etc.). Certain nodes apply to some uses and not others. For example, data standard models will not have a ***Models*** sub-node, and only integrated standards have a ***Standards*** sub-node and allow more than a single ***Models*** sub-node.

Model Changes

The only structural change is the addition of a ***Regulatory Authority Acceptance*** node. To be said at the outset, this is a nice-to-have. It is largely redundant to information present, for example, in FDA's Data Standards Catalog.

What it does do is provide CDISC the opportunity to address a common misperception in the industry that a RA's requirement start date of one version of a standard is the death knell of all prior versions for that RA. Regulatory authorities routinely require the use of multiple versions of a standard at the same time, meaning that any of the versions required within a given window fulfills the requirement. This is a critical understanding for organizations in planning migration and adoption of a version of a standard, and many refrain from up-versioning efforts, or feel compelled to meet an artificial time horizon as a result.

Content Changes

The other item to note is a conceptual/content change for ADaM. ADAM currently does not have a ***Classes*** node under the standard root. That will be discussed in detail in the next section.

Classes

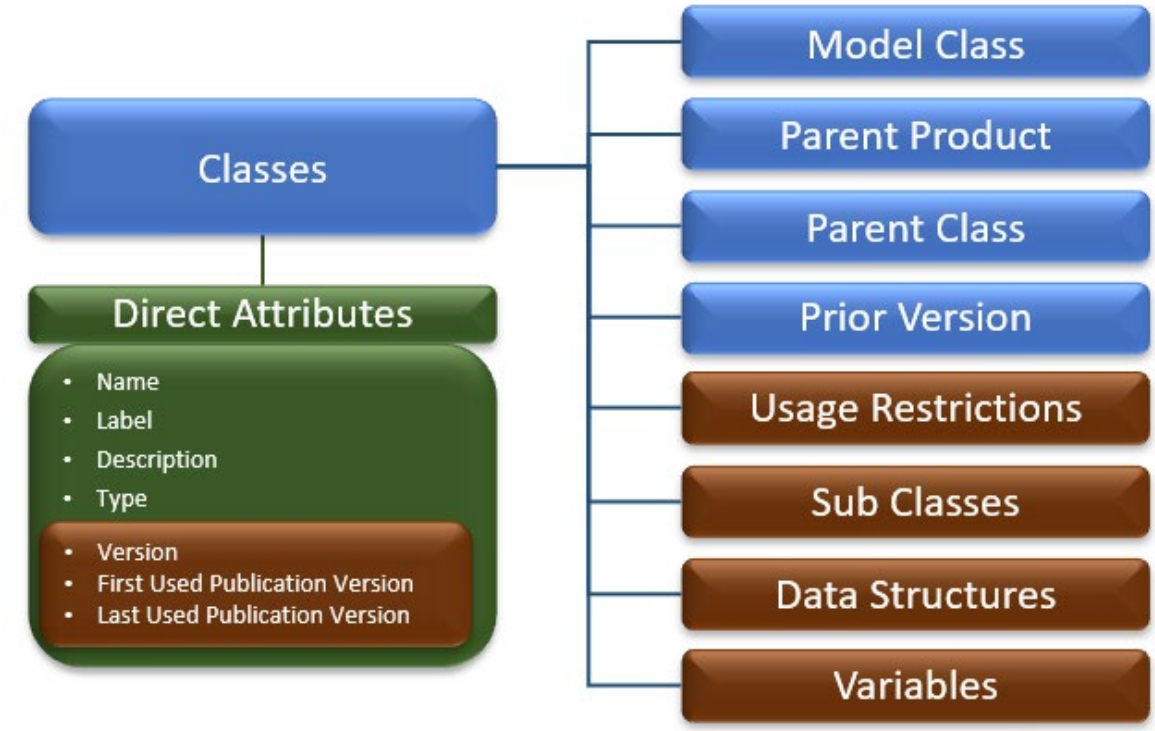

Figure 5. Classes Representation

Direct Attribute Changes

Several versioning attributes are added to the Class. To date, CDISC has maintained published versioning at the overall standard level, aligned to the published document. Organizations, however, have to manage change over time, and therefore establish bespoke mechanisms for tracking changes. In establishing these attributes, CDISC not only meaningfully simplifies the task of ingesting the standards into an organization's metadata reference, it facilitates tool development and adoption.

In that context, establish that a new version is required when:

- A change to a direct attribute occurs.
- A change in the linked content occurs. Given that this will reflect all class variables as well as data structures within that class, it generally guarantees each publication version of a class will be a distinct version to date. While this means that, practically, the versioning could be omitted, maintaining it consistently throughout the hierarchy has benefit for downstream consumption.

A primary assumption of this approach is that a change log is not needed, because CDISC provides a mechanism for generating one. While the Library UI is constrained to comparing just the latest versions, a general understanding from CDISC is that API calls can compare any two versions.

Model Changes

The new ***Usage Restrictions*** sub-node is a formalization of an existing concept currently implemented at the domain and variable level in some models, although largely as a semi-structured text field. This approach disambiguates the distinct restrictions and represents the concept as directly actionable data (see the later section on this node).

The new ***Sub Classes*** sub-node formalizes the concept of a sub-class across standards. To date, it occurs as an attribute of ADaM ***Data Structures*** and is not necessarily consistent in its meaning. The

proposal is to establish and adhere to a consistent definition of ***Sub Class*** (currently this does not appear to be in the CDISC Glossary), something like:

- *An extension of an existing Class that requires one or more variables not established by the existing Class to which it belongs.*

Content Changes

In SDTM and CDASH:

- This moves "Associated Persons" and "Findings About" from ***Class*** to ***Sub Class***, disambiguating the current content.
- "Findings About" becomes a ***Sub Class*** under "Findings".
- "Associated Persons" becomes a ***Sub Class*** under "Findings", "Events", "Interventions", as well as under "Special Purpose" but applicable to only the DM and CO domains (See ***Usage Restrictions*** under the ***Sub Class*** node).

For ADaM, it is more nuanced:

- The definition is consistent with the use of ***Sub Class*** in BDS/TIME-TO-EVENT, BDS/POPULATION PHARMACOKINETIC ANALYSIS, and potentially others.
- The ADaM OCCDS/ADVERSE EVENT ***Sub Class***, however, is similar to the CDASH concept of a Scenario, or to a domain/dataset implementation of a GOC Class in SDTM (e.g., AE). Resolving this will likely require re-defining ADVERSE EVENT from a ***Sub Class*** to something else in the hierarchy.

The ***Data Structures*** sub-node reflects a harmonized convention across CDASH, SDTM/SEND, and ADaM. Currently, this node is named ***Domains, Datasets,*** or ***Data Structures*** depending on the data standard, but in terms of operational use it is the same item. Consolidating on ***Data Structures*** as used in ADaM establishes a higher-order concept applicable to all standards, and has the added benefit of side-stepping the Dataset vs. Domain debate.

Variables is a harmonized node, consolidating the CDASH ***Model Fields*** and SDTM ***Class Variables*** nodes, capturing the class-level implementation of variables. Note that the parent context (model, class) has been removed. This is true throughout the model to reflect that variables have a harmonized definition. While there are certain attributes that will only be applicable to some data standards, the current approach has led to fragmented representation and operational challenges.

In addition, the following content changes are required to align all of the standards to this model:

- ADaM ***Class*** is currently a direct attribute of ***Data Structures*** and not present in the hierarchy. By all objective measures, Class is the same concept (with different allowed terms) across all standards, and it should be modeled as such.
- Disambiguate values in **Class** from **Sub Class** from **Variable Group** across standards (e.g., CDASH ***Class*** of "Associated Persons – Identifiers"). Diverging from the published term in the document should not be seen as an obstacle as long as the value in the Library is semantically aligned (e.g., a synonym).
- In SEND, SDTM, and Integrated models, convert "General Observation Class" from ***Class*** to a dataset-level direct attribute. Given there is already a "Type" attribute, this can be expanded to include "General Observation Class" in addition to the existing term of "Class", adding value to the attribute rather than making it simply redundant to the node name. Alternatively, a distinct attribute could be created. Operationally, the term of GOC is for ease of reference, and not truly established in the model hierarchy. The other option would be to establish a super-class node, but that provides little benefit while adding complexity to the model.
- Populate attribute/relationship in metadata if known at time of publication of standard (i.e., digitalize rather than digitize). For example, in the CDASH model representation, special-purpose datasets such as DM are not shown to be under the ***Special Purpose*** class.

Usage Restrictions

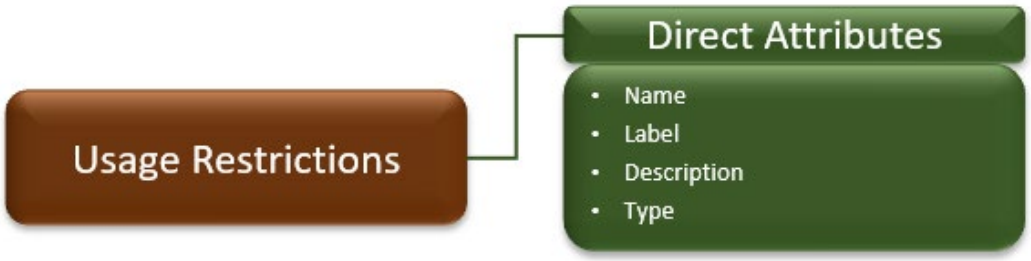

Figure 6. Usage Restrictions Representation

Model Changes

Usage Restrictions is introduced as a new node rather than a direct attribute to allow for multiple values and types to be specified. As noted above, the node is a formalization of an existing concept currently implemented for domains and variables. There is currently not an established use case for populating this at the ***Class*** level, but there is for ***Sub Class***, arguing for it to be consistently established. This results in the node occurring at multiple levels within the model, which is true for several of the nodes.

Type has the following proposed values:

- Restricted to Data Structure
- Prohibited from Data Structure
- Prohibited from Implementation Guide

See ***Sub Class, Data Structures,*** and ***Variables*** node sections for specific use cases.

The proposal is to have the name as a simple text attribute constrained to existing data-structure and IG names. It could be implemented as a link in the case of “Restricted to Data Structure”, but since the others are intended to disallow such relationships, text is a more generalized approach.

Sub Classes

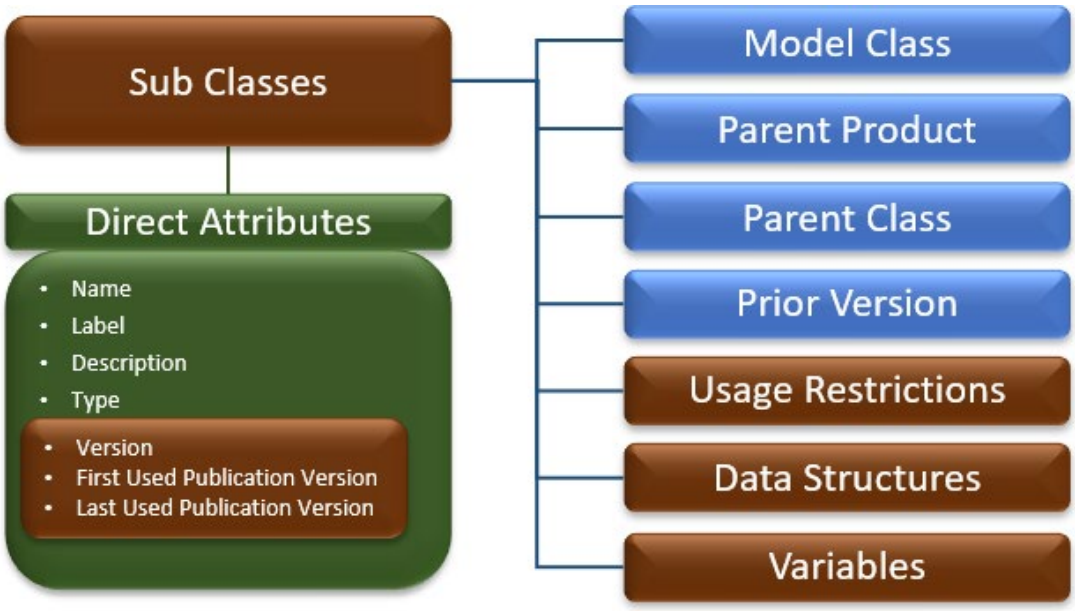

Figure 7. Sub Classes Representation

Direct Attribute Changes

The same version-related direct attributes as ***Class*** are represented. Similarly, any change to direct attributes or links establishes a new version. Again, there will a low incidence of a structure with no changes since the last version, but consistency and transparency drive automation.

Model Changes

Sub Class is a new node, largely the same as ***Class***, without the sub-node for subordinate classes. The "Type" attribute is "Sub Class" or "General Observation Sub Class".

As an example, the ***Usage Restrictions*** sub-node would be used by the "Associated Persons" ***Sub Class*** under the ***Special Purpose*** class to indicate that it is restricted to the "DM" and "CO" data structures in SDTM.

Data Structures

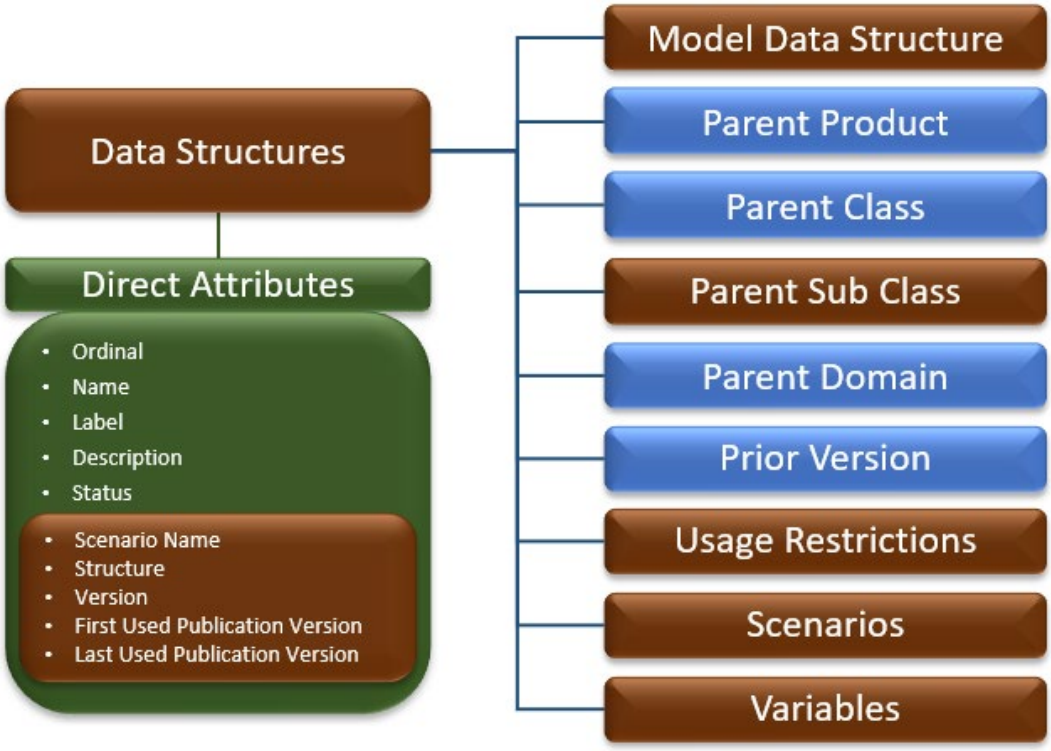

Figure 8. Data Structures Representation

Direct Attribute Changes

As noted under ***Class***, the node for ***Domains, Datasets, and Data Structures*** are harmonized to a single term across all standards, ***Data Structures***.

The same version-related direct attributes as ***Class*** are represented in ***Data Structures***. Similarly, any change to direct attributes or links establishes a new version. Again, there will a low incidence of a structure with no changes since the last version, but consistency and transparency drive automation.

The direct attribute “Scenario Name” is not necessarily new, but to date has only applied to CDASH. This promotes it to the broader model, as is the case with the ***Parent Domain*** link. Not represented, but for consideration is a direct attribute identifying data structure type. This may be helpful in distinguishing, for example, Scenarios (a specific implementation of a Domain) from general domains, but as it stands is not essential (i.e., a test for “Scenario Name” not null identifies Scenarios).

The final direct attribute to note is “Structure”. This is a harmonization of the “Dataset Structure” attribute in SEND and SDTM content, as well as the “Description” attribute in ADaM (see below).

Model Changes

The ***Model Data Structure*** sub-node has been harmonized to adhere to consistent nomenclature.

In addition to the parent ***Class***, a link is added to support ***Sub Class***.

The ***Usage Restrictions*** is at the data structure level primarily to identify SDTM vs. SEND usage. For example, the Subject Repro Stages (SJ) domain would have a “Prohibited from Implementation Guide” restriction for SDTMIG, something not readily apparent in the current Library.

Across the standards, variables are published in categories aligned to purpose (e.g., “Identifiers”, “Timing”, “Dictionary Coding Variables for MedDRA”), often as distinct tables leading to similar fragmentation in the modeling. These categories are sometimes captured in **Class** names. Sometimes,

they are in the direct attribute “Role” under variables, while in other instances, they are established as distinct nodes within the hierarchy.

Not only is this confusing for consumers, it is ultimately counterproductive, as will be covered under variables. To resolve this:

- In ADaM, demote ***Variable Sets*** to a direct attribute of the ***Variable*** node (e.g., “Role” or the new “Variable Group” attribute in SDTM).
- In CDASH, disambiguate the role content from Class and place at the variable level (e.g., “Role”).
- See the Variable section below for further details.

Content Changes

There are a number of other content realignments in the ***Data Structures*** node:

- In the CDASH content:
    - Under Scenarios, rename the “Domain” attribute to “Label” to be consistent with the other standards.
    - Under Scenarios, rename the “Domain Name” attribute to “Name” to be consistent with the other standards.
- In the SDTM and SEND content:
    - Evaluate the use of “Status”. It is only ever populated for SDTM and SEND, and seems redundant to the Standard ***Root*** attribute of “Registration Status”.
- In the ADaM content:
    - Rename the attribute “Description” to “Structure” (previously “Dataset Structure”) to be consistent with use in other standards. (the “Description” attribute name is already in use by SEND, CDASH, and SDTM, and the ADaM values are more aligned to “Structure”).
    - Harmonize ADaM metadata structure internal to the standard. For example, the content of the “Data Structure Name” attribute in the ***Data Structure*** node is inconsistent with the content of the “Data Structure Name” attribute in the ***Variable*** node (at least as represented in the Excel extract, the JSON is structured differently).
- In the Integrated model content:
    - Shift the “Status” attribute (already used in SEND and SDTM for publication state) content to establish a parent or other link to be consistent with other standards. This holds even if the SEND/SDTM Status attribute is deprecated.

Variables

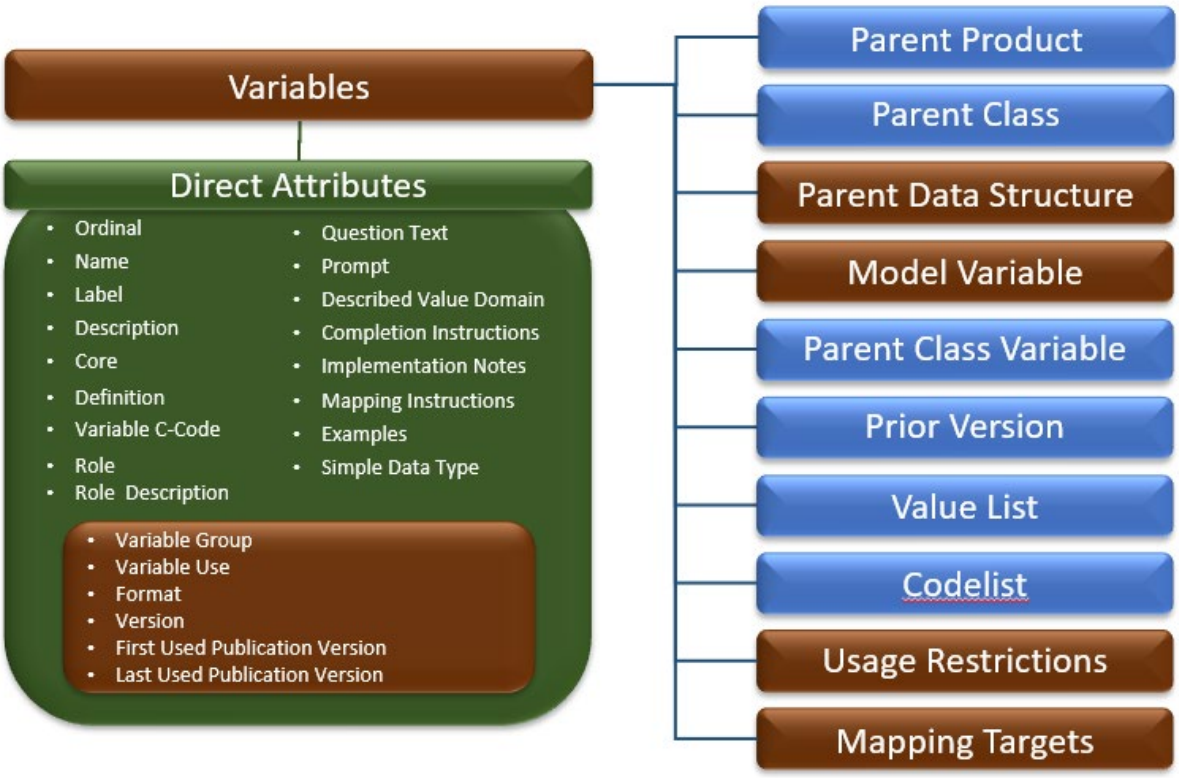

Figure 9. Variables Representation

The ***Variables*** node consolidates all data standards variables to a single, harmonized footprint.

Direct Attribute Changes

The same version-related direct attributes as ***Class*** are represented in ***Data Structures***. Similarly, any change to direct attributes or links establishes a new version. At the variable level is where organizations will find the most value of this versioning. Knowing which variables have no appreciable difference from the prior version is a fundamental first step in an impact analysis.

The new "Variable Group" and "Format" attributes are being driven by the latest SDTM data model, which is disambiguating previous attributes and establishing some new ones. The new "Variable Use" attribute categorizes the basis for including a variable in a ***Data Structure***, and is detailed under Content Changes below.

Model Changes

The ***Parent Data Structure*** and ***Model Variable*** sub-nodes are harmonized names of pre-existing nodes that were specific to a data standard.

As noted previously, the ***Usage Restrictions*** sub-node is a consolidation of various attributes. The specifics of the remapping are noted below under each standard as appropriate.

The ***Mapping Targets*** sub-node is a generalization of the CDASH link for SDTM Mapping Targets. The specifics of the link go beyond the scope of this paper, as they are dependent on the underlying architecture. That said, this is a fundamental gap in the broader modeling that limits end-to-end automation. Such links will allow support of source-to-target mapping between standards (e.g., CDASH to SDTM, SDTM to ADaM, ADaM to TFL).

Content Changes

The ***Variables*** node has perhaps the most significant change proposed to content. Currently, ***Data Structures*** tend to contain only those ***Variables*** that are explicitly listed in the source documents. For example, the "Associated Persons" and "Findings About" sub-classes represent just the new variables as a stand-alone fragment. In another example, a number of assumptions in the SDTMIG identify variables that are generally not used for a given domain. These variables are permissible (allowed per the model) but do not have an established use case that warrants their inclusion in the domain structure table in the IG. That does not mean that their use is prohibited.

Details on how to operationalize this content are present in the reference documents, but there is no actionable metadata. This has the following impacts, to name a few:

- There often is no published full representation of data structures. This requires subject-matter experts to assemble a domain's allowable variables for a given use case based on several inputs (e.g., the IG structure table, the model tables for class, exposition in reference documents), leading to bespoke solutions.
- Processes and technical solutions have no consistent framework for translating, up-versioning, and otherwise incorporating the published content into operational use, again leading to bespoke solutions.
- There is no objectively constant order for variables as published. Some standards require this while others don't, but regardless, it becomes an ongoing activity for implementers, where a variable might be ordinal 11 in one use case, and ordinal 13 in another. While a small detail, this is an ever-present drag on data integrations and content management.

The proposal is to published each ***Data Structure*** as a full and complete representation. For SDTM and SEND, this entails integrating the model permissible class- and role-related variables that are not currently published in the documents for a given domain. For ADaM, this entails integrating the ***Variable Sets*** into a single data structure for a given class and resolving how to represent them.

There are a host of other challenges as well, many detailed in this paper, that are much easier to state than resolve.

Even so, there is a fundamental disconnect between the published standards and their operational use. The benefits of operationalizing separate ADSL *Identifier, Subject Demographics,* and *Treatment* **Variable Sets** the way they are represented in the model are limited as best. Nearly universally, implementers create an ADSL reference, or a Demography collection form, or a local lab scenario for LB in SDTM, each with all the variables needed. In another example, Associated Persons is published as a variable set fragment. The proposed model would establish "Associated Persons" as a ***Sub Class*** of "Findings", as an example, and present a full, integrated reference for use.

This approach allows CDISC to explicitly identify variable usage, and would be the basis for the new direct attribute "Variable Use". While the terms for this attribute are largely based on SDTM/SEND conventions, they are intended to be broadly applicable:

- IG Specified – For variables listed in the IG for a given ***Data Structure***
- Model Permissible – For variables allowed per the model, but otherwise unconstrained.
- Generally Not Used – For variables identified in the IG as not having a known use case, but still allowed.

Additional terms, the use of which is likely more controversial, are:

- IG Prohibited – If used, for variables disallowed in a specific ***Data Structure*** use case. Whether variables such as AEOCCUR are simply omitted from the AE ***Data Structure*** or included with this category results in the same outcome: a metadata-driven understanding of each variable's validity.

- Model Prohibited – While this is required for operational use, it is not applicable to CDISC published content (the models define the variables). Establishing the term is in CDISC's interest, however, in promoting a consistent understanding by industry and regulatory authorities. An example of this is, for SDTM, the inclusion of SUBJID in subject-level datasets when there are multiple participations. FDA asks for this variable even though it is not allowed per the model. CDISC may also find use cases for it, such as in CDISC 360i, or in representing CORE Rules for regulatory authorities.

The combination of this attribute and the ***Usage Restrictions*** node provides a full accountability of use. It supports tooling and automation, and make clear from the outset certain conformance requirements that historically haven't been evaluated until datasets have already been produced.

Beyond this, there are a number of other content realignments in the ***Variables*** node:

- In CDASH
    - Disambiguate the role content from Class and place at the variable level (e.g., "Role").
    - Convert the "Domain Specific" direct attribute flag to the ***Usage Restrictions*** sub-node, by populating "Restricted to Data Structure" with the appropriate domain code.
    - The ***Fields*** node is harmonized to ***Variables***
- In SEND/SDTM
    - Rename "Notes" direct attribute to "Implementation Notes"
    - Disambiguate the semi-structured text in the "Usage Restrictions" direct attribute and map to the ***Usage Restrictions*** sub-node
- In ADaM
    - The ***Variable Sets*** node has been demoted to a direct attribute. The terms are a mix of "Role" and "Variable Group" as implemented in other standards. For historical alignment, it might be best to represent the terms under "Role", allowing the "Variable Group" to be developed in the future.

Lastly, the ***Value List*** and ***Codelist*** sub-nodes are not specified further as they aren't really changing from the current implementation. They both are required to be sub-nodes to support a one-to-many relationship from the ***Variable*** node to them.

TERMINOLOGY MODEL

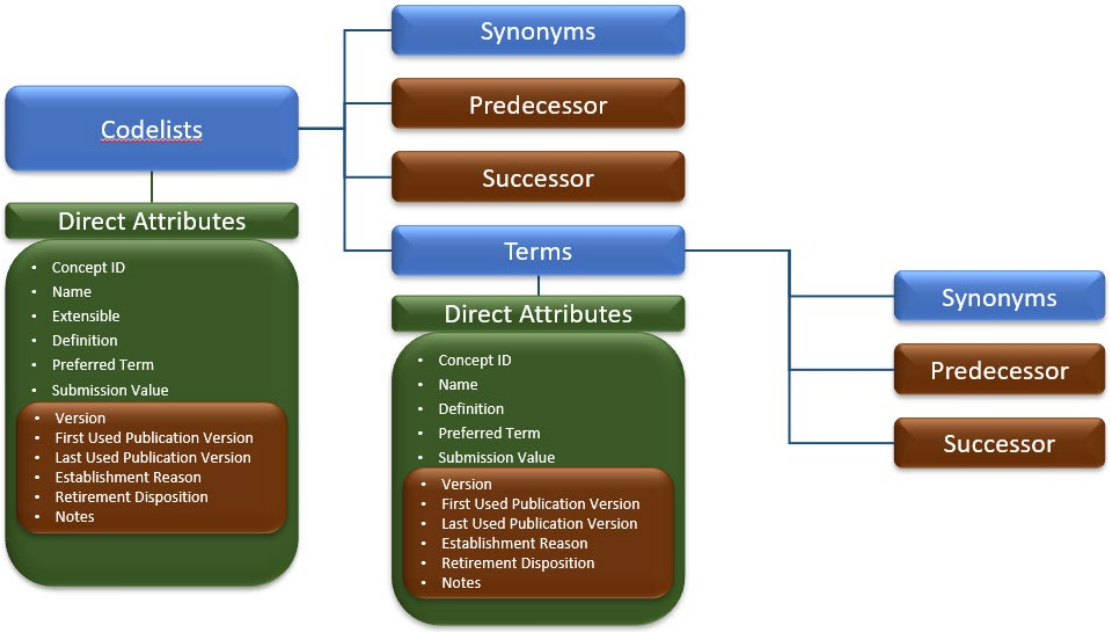

Figure 10. Terminology Model

Terminology is a much simpler model, and fundamentally has a single, if fundamental, gap: the support of managing terms over time. To address this, both Codelists and Terms modeling need to be expanded. CDISC is already doing much of the work, as it publishes each version through the change logs, but it is not, to date, incorporating that work directly into the metadata model.

Direct Attribute Changes

For both ***Codelists*** and ***Terms***, six new direct attributes are established. The first three align to the same versioning model as used by the data standards.

The attribute “Establishment Reason” identifies why a term was created, with the following values:

- New
- Replacement
- Split
- Join

The attribute “Retirement Disposition” identifies why a term was discontinued, with the following values:

- Removal
- Replacement
- Split
- Join

The attribute “Notes” captures any exposition related to the item.

The intent is to formalize content that CDISC develops in the course of terminology management, often published in presentations and change logs, but not represented directly in the Library content. The Replacement / Split / Join dispositions make clear how to traverse the versions of terminology (e.g., an Establishment Join indicates that a node will have multiple ***Predecessor*** nodes, while a Retirement Split indicates that it will have multiple ***Successor*** nodes).

These, in conjunction with the versioning attributes will fundamentally streamline how organizations are able to manage terminology, assess impact of changes, and up-version / down-version a terminology reference in operational use.

Model Changes

Two new sub-nodes are proposed for both ***Codelists*** and ***Terms: Predecessor*** and ***Successor***. These are links to existing content in other versions of the CT package, just as the data standards have. In contrast, however, these bi-directional links support traversals forward and back in time from any starting point.

The fundamental requirement is that these links have leverage an unambiguous, unique identifier. A simple solution is to publish a generated, single-value unique identifier for each codelist and term. Alternatively, for ***Codelists,*** the use of C-Code is sufficient, as each Codelist C-Code must be unique to a given package. For ***Terms***, the link must be to a specific term within a specific codelist. If there isn't an established unique identifier beyond C-Code, then the link must take into account identifiers for both the Codelist and Term.

These links, in conjunction with the direct attribute additions, provide essential tools for implementers. They connect terms over time and allow the building of a continuum of use for codelists and, more importantly, terms. With them, organizations will begin to be able to streamline and automate up-versioning and down-versioning of terminology, a requirement often occurring for integrated analyses. There may be more needed, but this is a start.

CONCLUSION

There's a lot of moving parts, and while many of the remodeling changes are supported by implementation experience (both good and bad), it takes more than a village to build a robust solution. Some of the proposals will resonate more than others in the industry. In short, this proposal is a start not a finish, intended to put forth the ideas into the broader marketplace.

In addition, changes of this scope will require CDISC to exert discipline on its standards teams – not always an easy task. As can be seen from viewing all of the data standard models as a whole, a walking, talking duck may be called a mallard by some and a water fowl by others. It may require the concerted efforts of the broader community (e.g., the CDISC Advisory Board, internal leadership, and internal groups along the lines of the Global Governance Group and Conformance Rules Operational Governance) to find the solution. Longterm, as the models harmonized and adopt aligned terminology, this should feed back to the standard documents themselves, harmonizing on terms and concepts, and making it easier for people and systems to understand.

What can be said definitively is that each inconsistency, each hiccup in modeling is a drag on optimization, leading to standard-specific implementations and a fragmented ecosystem of tools and processes. It also makes it that much more challenging to build out a semantic layer, complicating efforts like CDISC 360i, and hindering the ability of AI systems to reliably and accurately engage with the standards. Whatever a true end-to-end solution may look like, the model should facilitate rather than be a roadblock.

Further, a number of CDISC initiatives are building out additional models supporting implementation of standards. Looking at terminology, CDISC is currently publishing supporting references such as paired lists, code tables, etc. These could be understood in some cases as extensions of the terminology model and in others as linkages and integrations between terminology and data standards. Additionally, the Unified Study Definitions Model (USDM) builds upon much of this content, with direct connections to terminology and study design concepts developed in the data standards, and the Analysis Results Standard establishes linkages to ADaM datasets as inputs.

In other words, CDISC is building a unified metamodel from the bottom up.

Pegasus's well at Hippocrene may be a source of inspiration (literally "to breath in"), but it takes constant effort to bring magic into the world. If we aim to have a true end-to-end solution, now seems an opportune time to look from on high and leverage opportunities we might otherwise miss.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Carlo Radovsky
Immanant
Carlo.Radovsky@Immanant.com
www.immanant.com